

Aula 4

Devops e Integração Contínua

Prof. Mauricio Antonio Ferste

1

Conversa Inicial

2

Docker

- Indicação de livros
 - Containers com Docker – Do desenvolvimento à produção
 - Autor – Daniel Romero
 - Docker para Desenvolvedores
 - Autor – Rafael Gomes

3

Máquinas virtuais e sua utilidade em Devops

4

Máquinas virtuais

- Máquinas virtuais de Tipo 1 (Bare Metal):
Essas VMs são executadas diretamente sobre o hardware, sem a necessidade de um sistema operacional hospedeiro. O hipervisor age como um sistema operacional dedicado para gerenciar a execução das VMs. Isso resulta em melhor desempenho e eficiência, sendo comumente utilizado em ambientes de produção e servidores

5

Máquinas virtuais

- Máquinas virtuais de Tipo 2 (Hospedadas):
Essas VMs são executadas sobre um sistema operacional hospedeiro convencional. Um aplicativo de virtualização é responsável por gerenciar as VMs, e o sistema operacional hospedeiro executa tarefas essenciais. (...)

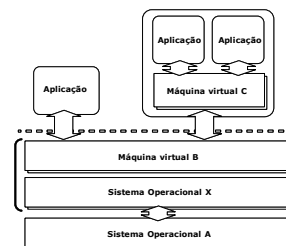
6

Máquinas virtuais

- (...) Embora sejam mais fáceis de configurar e usar, as VMs de Tipo 2 geralmente tem um desempenho inferior em comparação com as de Tipo 1 e são mais adequadas para ambientes de desenvolvimento e testes

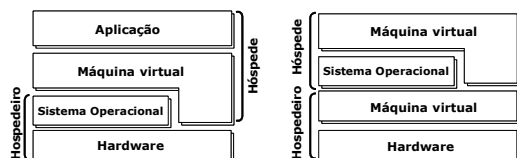
7

Máquinas virtuais



8

Máquinas virtuais



9

Containers: introdução ao Docker

10

Contêiner

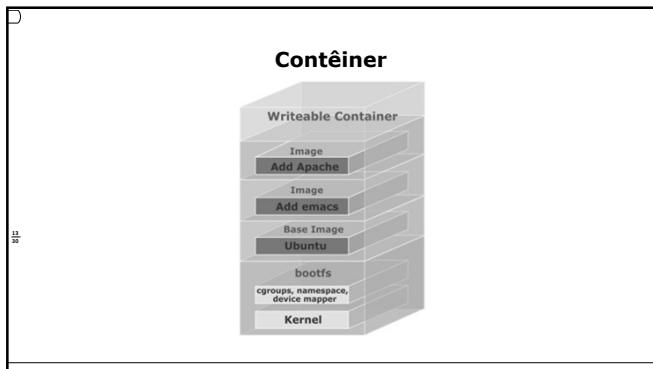
- Contêineres são ambientes leves e isolados que encapsulam uma aplicação e suas dependências. Eles proporcionam consistência entre diferentes ambientes de desenvolvimento, teste e produção, eliminando as divergências que podem surgir devido a diferenças nos sistemas operacionais ou configurações

11

Contêiner

- Um contêiner representa um ambiente isolado para o seu código, caracterizado pela ausência de conhecimento sobre o sistema operacional ou os arquivos do host

12

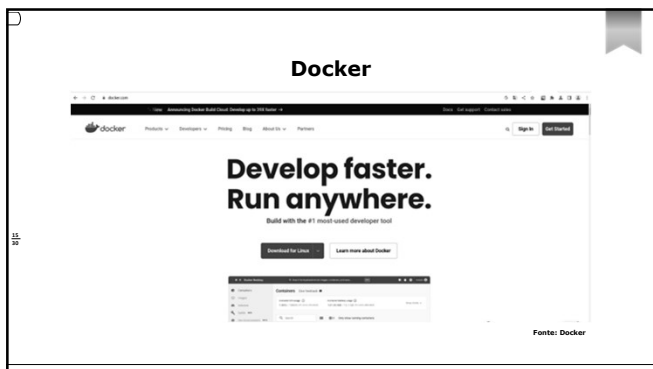


13

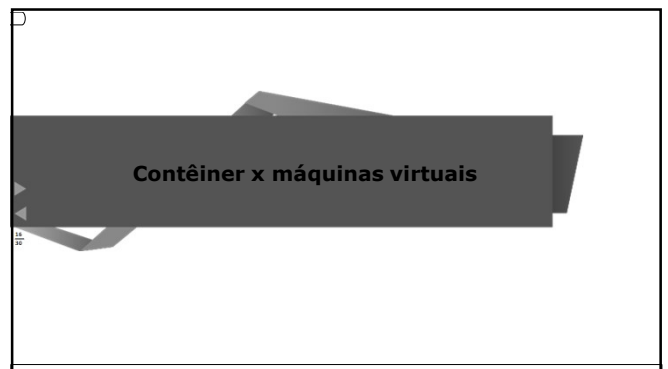
Docker – Definição

- Uma imagem do Docker consiste em sistemas de arquivos sobrepostos, sendo a base um sistema de arquivos de inicialização, chamado bootfs, semelhante ao típico sistema de arquivos de inicialização do Linux/Unix. Normalmente, os usuários do Docker não interagem diretamente com o sistema de arquivos de inicialização. Após a inicialização de um contêiner, ele é movido para a memória e o sistema de arquivos de inicialização é desmontado para liberar a RAM usada pela imagem do disco (Docker, 2019)

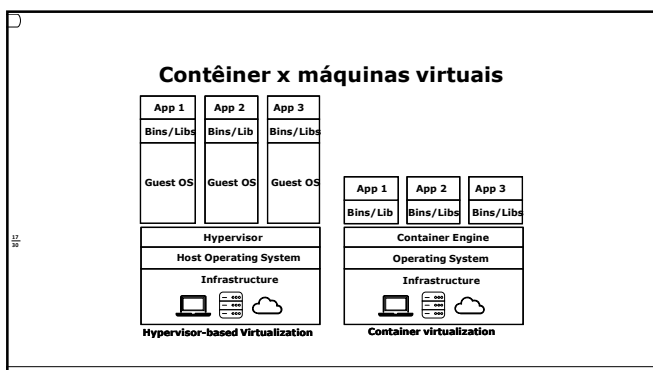
14



15



16



17

Vantagens e desvantagens

- Os contêineres proporcionam mais agilidade
- Os contêineres economizam no licenciamento de VM

18

Vantagens e desvantagens

- Os contêineres permitem adoção híbrida e multicloud

vedpku / Shutterstock
Raif Liebhoid / Shutterstock
DigitalOcean / CC-0P
tanuha2001 / Shutterstock
Burdun Ilya / Shutterstock

19

Vantagens e desvantagens

- Os contêineres se integram a sistemas legados

20

Orquestração com Kubernetes

21

YAML

```
YAML: 1.2
YAML: Y Ain't Markup Language

What is it:
YAML is a human-friendly data serialization
language for all programming languages.

YAML Resources:
YAML Specifications:
- YAML 1.2
- Revision 1.2.2 # Oct 1, 2021 "New"
- Revision 1.2.1 # Oct 1, 2009
- Revision 1.1.3 # Jul 23, 2009
- YAML 1.1
- YAML 1.0

YAML Matrix Chat: #chat.yaml.io # Our New Group Chat Room!
YAML IRC Channel: libera.chat.yaml # The old chat
YAML Mailing List: yaml-core # Obsolete, but historical

YAML on GitHub: # github.com/yaml/
YAML Specs: yaml-spec/
YAML 1.2 Grammar: yaml-grammar/
YAML Test Suite: yaml-test-suite/
YAML Issues: issues/

https://yaml.org/
```

22

Kubernetes

23

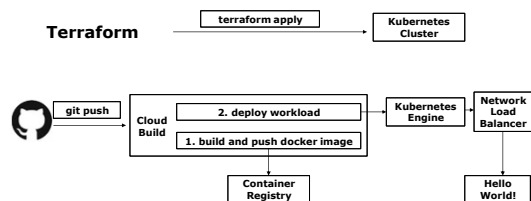
Orquestração com Kubernetes

24

Escalabilidade e distribuição com contêineres

25

Contextualização IAC



26

Vantagens e desvantagens

- Com a automatização do provisionamento e da configuração de recursos, a infraestrutura pode ser criada, modificada ou removida rapidamente, reduzindo o tempo de lançamento de novos produtos e serviços
- Redução de erros: A padronização e a consistência do código, aliadas aos testes automatizados, diminuem significativamente a quantidade de erros humanos e a probabilidade de falhas na infraestrutura

27

Vantagens e desvantagens

- Versionamento e rastreabilidade: O uso de sistemas de controle de versão, como Git, permite rastrear mudanças, identificar problemas e reverter para versões anteriores, facilitando a manutenção e a auditoria
- Escalabilidade e flexibilidade: A automação e a padronização proporcionadas pelo Terraform e IaC facilitam a escalabilidade de recursos e a adaptação a diferentes ambientes, diminuindo a necessidade de intervenção manual e possibilitando o uso de infraestrutura imutável

28

Vantagens e desvantagens

- Redução de custos: A implementação de Terraform e IaC permite otimizar o uso de recursos, identificar e eliminar desperdícios, e ajustar a infraestrutura de acordo com as necessidades específicas do negócio. Isso resulta em economia de custos e maior eficiência operacional

29

Vantagens e desvantagens

- Segurança e conformidade: A padronização do código e a integração com políticas de segurança garantem a conformidade com as normas e regulamentações. Além disso, o uso de IaC permite a criação de ambientes isolados, que facilitam a detecção e a correção de vulnerabilidades

30

